Epoch 1/20

1682/1682 [==============================] - ETA: 0s - loss: 15.9361 - accuracy: 0.52212024-04-19 04:17:23.016617: W tensorflow/tsl/framework/cpu\_allocator\_impl.cc:83] Allocation of 10509312000 exceeds 10% of free system memory.

2024-04-19 04:17:32.830977: W tensorflow/tsl/framework/cpu\_allocator\_impl.cc:83] Allocation of 1847623680 exceeds 10% of free system memory.

2024-04-19 04:17:33.819207: W tensorflow/tsl/framework/cpu\_allocator\_impl.cc:83] Allocation of 2463498240 exceeds 10% of free system memory.

1682/1682 [==============================] - 1085s 643ms/step - loss: 15.9361 - accuracy: 0.5221 - val\_loss: 9.5351 - val\_accuracy: 0.6304 - lr: 1.0000e-05

Epoch 2/20

1682/1682 [==============================] - ETA: 0s - loss: 6.0139 - accuracy: 0.66002024-04-19 04:35:20.282093: W tensorflow/tsl/framework/cpu\_allocator\_impl.cc:83] Allocation of 10509312000 exceeds 10% of free system memory.

2024-04-19 04:35:26.363974: W tensorflow/tsl/framework/cpu\_allocator\_impl.cc:83] Allocation of 1847623680 exceeds 10% of free system memory.

1682/1682 [==============================] - 1074s 639ms/step - loss: 6.0139 - accuracy: 0.6600 - val\_loss: 3.4422 - val\_accuracy: 0.6782 - lr: 1.0000e-05

Epoch 3/20

1682/1682 [==============================] - 874s 520ms/step - loss: 2.1857 - accuracy: 0.7222 - val\_loss: 1.3856 - val\_accuracy: 0.6646 - lr: 1.0000e-05

Epoch 4/20

1682/1682 [==============================] - 770s 458ms/step - loss: 0.9544 - accuracy: 0.7631 - val\_loss: 0.8035 - val\_accuracy: 0.6565 - lr: 1.0000e-05

Epoch 5/20

1682/1682 [==============================] - 770s 458ms/step - loss: 0.5775 - accuracy: 0.7894 - val\_loss: 0.6579 - val\_accuracy: 0.6538 - lr: 1.0000e-05

Epoch 6/20

1682/1682 [==============================] - 772s 459ms/step - loss: 0.4544 - accuracy: 0.8073 - val\_loss: 0.6414 - val\_accuracy: 0.6454 - lr: 1.0000e-05

Epoch 7/20

1682/1682 [==============================] - 771s 458ms/step - loss: 0.4080 - accuracy: 0.8208 - val\_loss: 0.6640 - val\_accuracy: 0.6362 - lr: 1.0000e-05

Epoch 8/20

1682/1682 [==============================] - 771s 458ms/step - loss: 0.3844 - accuracy: 0.8292 - val\_loss: 0.6936 - val\_accuracy: 0.6368 - lr: 1.0000e-05

Epoch 9/20

1682/1682 [==============================] - 773s 460ms/step - loss: 0.3672 - accuracy: 0.8370 - val\_loss: 0.7280 - val\_accuracy: 0.6319 - lr: 1.0000e-05

Epoch 10/20

1682/1682 [==============================] - 773s 460ms/step - loss: 0.3529 - accuracy: 0.8428 - val\_loss: 0.7671 - val\_accuracy: 0.6303 - lr: 1.0000e-05

Epoch 11/20

1682/1682 [==============================] - ETA: 0s - loss: 0.3408 - accuracy: 0.8491

Epoch 11: ReduceLROnPlateau reducing learning rate to 5.999999848427251e-06.

1682/1682 [==============================] - 772s 459ms/step - loss: 0.3408 - accuracy: 0.8491 - val\_loss: 0.8006 - val\_accuracy: 0.6262 - lr: 1.0000e-05

2883/2883 [==============================] - 30s 10ms/step

Evaluation Metrics:

Precision: 0.6562

Recall: 0.6013

F1 Score: 0.6276

ROC AUC: 0.7549

PR AUC: 0.7933

MCC: 0.2874

y\_pred = (test\_scores > 0.6).astype(int)

...: y\_true = y\_test.astype(int)

...:

...: # Calculate evaluation metrics

...: precision = precision\_score(y\_true, y\_pred)

...: recall = recall\_score(y\_true, y\_pred)

...: f1 = f1\_score(y\_true, y\_pred)

...: roc\_auc = roc\_auc\_score(y\_true, test\_scores)

...: pr\_auc = average\_precision\_score(y\_true, test\_scores)

...: mcc = matthews\_corrcoef(y\_true, y\_pred)

...:

...: # Confusion Matrix

...: conf\_matrix = confusion\_matrix(y\_true, y\_pred)

...:

...: # Display evaluation metrics

...: print("\nEvaluation Metrics:")

...: print(f'Precision: {precision:.4f}')

...: print(f'Recall: {recall:.4f}')

...: print(f'F1 Score: {f1:.4f}')

...: print(f'ROC AUC: {roc\_auc:.4f}')

...: print(f'PR AUC: {pr\_auc:.4f}')

...: print(f'MCC: {mcc:.4f}')

Evaluation Metrics:

Precision: 0.7034

Recall: 0.5384

F1 Score: 0.6099

ROC AUC: 0.7549

PR AUC: 0.7933

MCC: 0.3204

In [5]: y\_pred = (test\_scores > 0.7).astype(int)

...: y\_true = y\_test.astype(int)

...:

...: # Calculate evaluation metrics

...: precision = precision\_score(y\_true, y\_pred)

...: recall = recall\_score(y\_true, y\_pred)

...: f1 = f1\_score(y\_true, y\_pred)

...: roc\_auc = roc\_auc\_score(y\_true, test\_scores)

...: pr\_auc = average\_precision\_score(y\_true, test\_scores)

...: mcc = matthews\_corrcoef(y\_true, y\_pred)

...:

...: # Confusion Matrix

...: conf\_matrix = confusion\_matrix(y\_true, y\_pred)

...:

...: # Display evaluation metrics

...: print("\nEvaluation Metrics:")

...: print(f'Precision: {precision:.4f}')

...: print(f'Recall: {recall:.4f}')

...: print(f'F1 Score: {f1:.4f}')

...: print(f'ROC AUC: {roc\_auc:.4f}')

...: print(f'PR AUC: {pr\_auc:.4f}')

...: print(f'MCC: {mcc:.4f}')

Evaluation Metrics:

Precision: 0.7659

Recall: 0.4836

F1 Score: 0.5929

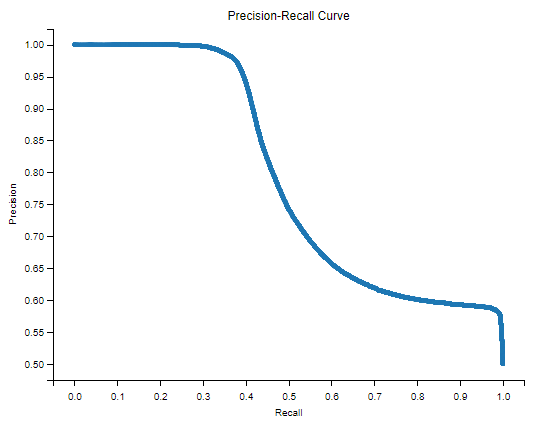
ROC AUC: 0.7549

PR AUC: 0.7933

MCC: 0.3613

Εικόνα που περιέχει κείμενο, διάγραμμα, γράφημα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα



Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, ορθογώνιο παραλληλόγραμμο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, γραμμή, διάγραμμα, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα